BIO-INSPIRED COMPUTING FOR OUTLIER DETECTION: SELECT STUDIES IN WEB 3.0 DOMAIN

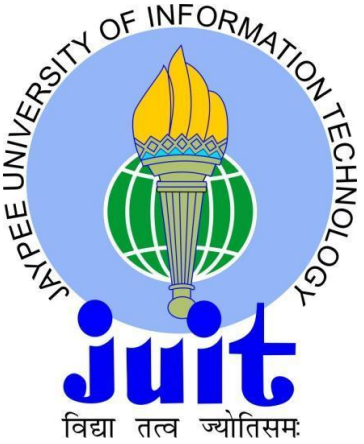
Synopsis submitted in fulfillment of the requirement for the Degree of

DOCTOR OF PHILOSOPHY

BY

REEMA ASWANI

Enrollment No. 136213



Under the supervision of

PROF. S. P. GHRERA

DR. SATISH CHANDRA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN-173234, HIMACHAL PRADESH, INDIA

FEBRUARY, 2018

TABLE OF CONTENTS

1. INTRODUCTION **3-5**

1.1 Problem Statement 3

1.2 Motivation and Contribution 4

1.3 Research Problems 5

2. OUTLIER DETECTION IN SUPERVISED SCENARIO **6-14**

2.1 Proposing Hybrid Algorithm for Outlier Detection 6

2.1.1 Grey Wolf Optimization 7

2.2 Outlier Detection among Influencer Blogs 10

2.2.1 Wolf search Algorithm 12

2.2.2 Bat Algorithm 13

3. OUTLIER DETECTION IN SUPERVISED SCENARIO **15-29**

3.1 Identifying Buzz in Social Media 15

3.1.1 Artificial Bee Colony 17

3.2 Detecting Fake Profiles on Social Media 21

3.2.1 Grey Wolf Optimization 23

3.2.2 Artificial Bee Colony 25

4. INTEGRATING CHAOS IN OUTLIER DETECTION **30-40**

4.1 Segregating Popular Online Content 31

4.1.1 Chaotic Cuckoo Search Algorithm 32

4.2 Identifying Spam Websites in Search Engines 35

4.2.1 Chaotic Firefly Algorithm 37

5. CONCLUSION & FUTURE SCOPE **41-42**

REFERENCES **43**

PUBLICATIONS **50**

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Progress in information, communication and multimedia technologies and the increasing expansion and use of the internet, intranets, extranets, web sites, etc., are generating gradual innovation in diverse areas, leading to the proliferation of new business styles based on information and knowledge (Garrigos, 2010), where the importance of networks, partnerships and alliances between firms and other agents is crucial. New networks and the advances in so-called Web 3.0 technologies are changing firm structures and value chains or value networks, and the configuration of decision-making processes for managers. Their efficient use is therefore crucial in the modern social and business environment in order to create and consolidate the competitive advantages of modern-day businesses (Garrigos-Simon et al., 2012).

Web 2.0 has several dominant applications including Flickr, Wikipedia, Facebook, MySpace, and YouTube that found ways to deliver the functionality of the "read-write Web" to large user groups, and to great effect. Plenty of money is left to be made in Web 2.0, but these marquee applications dominate the market. With Web 3.0, on the other hand, the explosion of data on the Web has emerged as a new problem space, and the game-changing applications of this next generation of technology have yet to be developed (Hendler, 2009).The domain of Web 3.0 has thus gained immense popularity with increased use of internet by individuals and organizations. The usage in the current scenario is not restricted to mere communication but targets various aspects of digital marketing. This engagement and interaction on platforms generates large amount of User Generated Content (both structured and unstructured).

On the other hand, Outlier detection is a widely explored area in various domains including wireless sensor networks (Zhang et al., 2010), medical data (Laurikkala et al., 2000; Gebremeskel et al., 2016) and fraud detection (Bolton and Hand, 2001) to name a few. Literature

highlights various outlier detection techniques and extensive reviews on methodologies (Hodge and Austin, 2004; Zimek et al., 2012). These approaches are categorized into nearest neighbor, statistical, clustering and classification based to name a few (Chandol et al., 2009).

K-Means and k-nearest neighbors are one of the most popular algorithms that can be used for clustering and classification datasets that is often used for outlier. The only pitfall of this approach is that it often falls into local optima. Further, due to increased variety and volume of the data, it has become computationally complex to analyze it using simple machine learning approaches. Thus, due to these issues in heuristic approaches, meta-heuristic approaches have become prominent over the decades. The current work thus attempts identify outliers (distinctly different data points) of various forms in the domain of Web 3.0 using bio inspired computing techniques.

1.2 Motivation and Contribution

The traditional machine learning algorithms have been extensively used for outlier detection. However, these algorithms are often prone to fall in local optima. Bio inspired computing algorithms are known overcome local optima and are popular for converging to a globally optimum solution. Existing studies in literature highlight that there are no studies in outlier detection using bio inspired computing approaches. Further, with the era of digitization and the advent of Web 3.0 there is increased use of social media and search engines. This results in generation of large amount of data both structured and non-structured including UGC.

The data contains critical information which can have implications in the domains of e-commerce, public policy, search engine marketing, influencer marketing to name a few. This data if mined and analyzed properly can give useful insights to be used in the above mentioned domains. Thus, the contribution of this work is twofold. The work targets select studies in the Web 3.0 domain highlighting outliers of different types. Methodologically, the work proposes several hybrid bio inspired computing algorithms by integrating them with traditional k-Means and k-nearest neighbors algorithms. The select studies thus use the proposed hybrid bio inspired approaches for outlier detection in relevant studies of Web 3.0.

The contribution in the work is both methodological as well as application oriented focusing on the domain of Web 3.0. Focused on three research problems in the Web 3.0 domain including search engine marketing, social media marketing and influencer marketing. The use of hybrid bio inspired computing algorithms eliminates locally optimum solutions and catalyze the convergence of the solution. The findings of the study have policy implications in domains of e-commerce, social media and influencer marketing.

1.3 Research Problems

Bio-inspired computing algorithms are known for optimizing an objective function either for minimization or maximization. These meta-heuristic approaches when integrated with traditional machine learning algorithms help in avoiding locally optimal solutions, expedite the convergence to the solution and enhance the accuracy of the obtained solution. In addition to the methodological contribution, the work addresses concerns in the domain of Web 3.0 surrounding outliers. To address the concerns surrounding detection/identification of outliers in the Web 3.0 domain the study proposes three research objectives with sub-problems having practical implications in the industry.

Problem I: Outlier Detection in Supervised Scenario

Problem II: Outlier Detection in Unsupervised Scenario

Problem III: Integrating Chaos in Outlier Detection

The research objectives attempt to address practical problems faced by organizations and individuals with the advent of Web 3.0 and increased use of internet in every facet of communication and engagement. The findings of the study have policy implications in domains of e-commerce, social media and influencer marketing.

CHAPTER 2

OUTLIER DETECTION IN SUPERVISED SCENARIO

This chapter explores the use of hybrid bio-inspired computing algorithms for detecting outliers in the supervised scenario. The supervised scenario may comprise of classification or regression datasets where in we have a continuous or nominal output label. The motivation behind these studies is to avoid locally optimum solutions and minimize convergence iterations which are significantly high when it comes to traditional machine learning approaches implemented for big datasets. The sections implements hybrid grey wolf optimizer, wolf search algorithm and bat algorithm. The latter have been explored for detecting outliers among influencer blogs.

2.1 Proposing Hybrid Algorithm for Outlier Detection

Traditional machine learning algorithms specially k-nearest neighbors has been a popular choice when it comes to outlier detection. It is a simple yet effective approach to adopt. The only fallback is that such approaches tend to fall in a local optima and often take a lot of time to converge for large datasets. With the exponential increase in the amount data being generated from different sources and the need to mine important and useful information from it. Data mining is becoming an integral part of any business. However, with the amount of data under consideration such approaches are becoming challenging when used in the real life scenarios with different data distributions and quantity of data being dealt with.

Meta-heuristics on the other hand specifically bio-inspired algorithms are known for optimizing an objective function and converging to a globally optimum solution with faster convergence. Outlier detection approaches have been popular but have rarely been explored through this lens of bio-inspired computing. The proposed approach uses a combination of the concept of grey wolf optimization (Mirjalili et al., 2014) and k-Nearest Neighbor algorithm. The discussion about optimization problems always includes an objective function where the motive is to optimize the given objective function. Thus, to apply any optimization technique the first step is to identify and select an objective function.

2.1.1 Grey Wolf Optimization

In the discussed problem of anomaly detection, for the proposed approach of Distance Based Grey Wolf Optimization for anomaly detection, the objective function is the accuracy of k Nearest Neighbor Algorithm. The thesis uses a modified version of GWO where the positions of search agents Xα, Xβ, and Xδ are not updated in every iteration. The α value i.e. the best positions for the wolves is used and is obtained from the kNN algorithm, β, δ and ω become the second, third and rest of the positions respectively. Fig,1 represents the pseudo-code for the proposed approach.

Pseudo-code for Modified GWO using kNN Algorithm

```
Begin
Initialize the grey wolf population
Initialize Test Data as Number of grey wolves (NumGW)
for each grey wolf NumGW
      /*calculate Accuracy and Position of the grey wolf */
/* i=1 Iteration 1 */
      R ← random number (between 1 to NumGW) is generated
      R is checked from a database of numbers so that R should not repeat
      /* Take data points one by one from TestData as position*/
Position ← TestData(R)
      Pass Position to kNN as input and get kNN Output
      Accuracy ← Target Value – kNNOutput
      Best_Value ← Acuuracy
      Best_Position ← Position
/* i=2 Iteration 2 */
for each grey wolf NumGW
      if (Best_Value > Acuuracy)
      then
            Best_Value=Accuracy
            Best_Position=Position
      End
End
```

Fig 1. Pseudo-code for Modified GWO using kNN Algorithm

The calculation done is the fitness of Xα which is the fitness for the best search agent. Again over an iteration, the best score and best positions i.e. best 'n' independent variables are found, where 'n' is the number of attributes in that dataset. The reason behind this is that in the proposed approach, α is pre calculated using the kNN by optimizing its accuracy and this makes the β, δ and ω solutions insignificant in this context.

Since, kNN is to find the best set of values (position of wolves) for anomaly detection and a data point of ‘n’ dimensionality, i.e. a row with ‘n’ columns as the (Target- Output) which is the minimum value is obtained. Thus, this becomes a minimization problem where the objective function for the optimization problem is to minimize the value of Target- Output or simply the kNN accuracy. The best set of values that we get as output from the kNN becomes the α positions for the GWO algorithm where the prey is hunt or simple where the best accuracy of kNN is achieved. The detailed approach is described in the flowchart (Fig, 2)

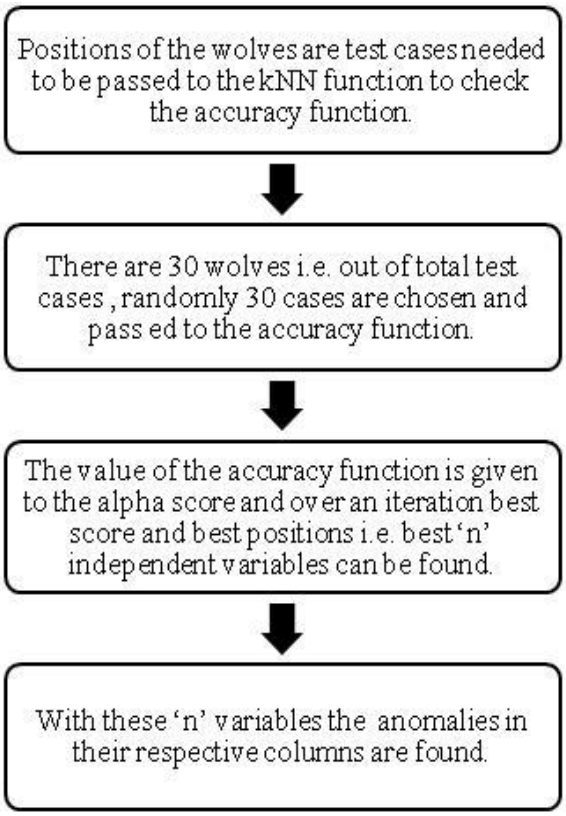

Positions of the wolves are test cases needed to be passed to the kNN function to check the accuracy function.
There are 30 wolves i.e. out of total test cases , randomly 30 cases are chosen and pass ed to the accuracy function.
The value of the accuracy function is given to the alpha score and over an iteration best score and best positions i.e. best ‘n’ independent variables can be found.
With these ‘n’ variables the anomalies in their respective columns are found.

Fig 2. Flowchart for GWO-kNN

The algorithm was tested on two supervised datasets namely Iris and Abalone for classification and regression respectively. The results for the accuracy for varying threshold values for outliers are depicted in Table 1.

Table 1. Results for GWO-kNN

Dataset	***Threshold**	**GWO-kNN Accuracy**

Iris	0.65	96.67
	0.75	80.00
	0.85	56.67
Abalone	0.5	90.0
	0.65	53.33
	0.75	26.67

*Threshold is a numeric value for every attribute, beyond which the data point is considered as an outlier

Further, the comparison with existing approaches surrounding the Iris dataset are illustrated in Table 2.

Table 2. Comparison of Classification Error Rate post Identification of Outliers

Approach	**Algorithm**	**Original Sample**	**Original Sample without outliers**
Existing Approaches (Acuna & Rodriguez, 2004)	LDA	31.82	26.23
	kNN (k=7)	31.55	27.65
	Rpart	31.86	33.24
Proposed Approach	GWO-kNN (k=1)	31.01	25.48
	GWO-kNN (k=5)	30.56	28.79
	GWO-kNN (k=10)	30.48	24.67

When compared with other existing approaches for outliers in a single class for Iris (Iris Virginica), it is seen that GWO-kNN over powers the rest in terms in running time. While the precision and recall is same for flock based algorithm and GWO-kNN. Table 3 illustrates the results for the class Iris Virginica.

Table 3. Results for class Iris Virginica

Approach	**Algorithm**	**Outliers Detected**	**Time (seconds)**	**Precision**	**Recall**
Existing (Murugavel & Punithavalli, 2011; Forestiero, 2017; Alam et al.,	PAM	7	8.56	N/A	N/A
	CLARA	7	8.55	N/A	N/A
	CLARAN	9	5.91	N/A	N/A
	Flock Based	10	N/A	**1.00**	**0.80**

2010)	Algorithm				
	HPSO-Clustering	N/A	6.00	1.00	0.25
Proposed	GWO-kNN	10	**5.45**	**1.00**	**0.80**

2.2 Outlier Detection among Influencer Blogs

This section of the work proposes a hybrid algorithm for detecting outliers in influencer blogs for the purpose of digital marketing. This study uses a mixed research methodology where in the data collected surrounding the website KPIs for 2751 influencer blogs on unique domains. A statistical t-test is conducted on the normalized data for the two sets of influencer web domains, with low and high spam score. Further, the significant metrics are used as KPIs for analyzing whether the influencer is spam or not using bio inspired optimization approaches integrated with K-Means for mining outliers. The subsequent sub-sections highlight detailed discussions surrounding the analysis.

The data is extracted through an API from the SEO Rank website that provides a holistic list of selected metrics provided by various data providers like Majestic, Ahref, Moz, SemRush and Webmaster tools. These data providers have developed ranking mechanisms that are used worldwide for identifying the position of a page in organic search. Table 4 provides the holistic list of metrics extracted.

Table 4. List of Metrics for Influencer Blogs

	Data Provider	**Metric**	**Description**
1.	Moz	Domain Authority	Prediction of the ranking of domain on search engines. Depends on links, Moz Rank and other metrics.
2.		Page Authority	Prediction of how a given URL may be ranked on search engines, associated with number of links, Moz Rank, and others.
3.		Moz Rank	Link popularity score indicative of importance of the page on the web.
4.		Moz Trust	Link trust checks for links from trustworthy sources.

5.		Links In	Links to the web page, includes equity, or non-equity both internal and external links.
6.		External Equity Links	Number of external equity links to the URL
7.		Spam Score	Based on number of sites penalized (de-listed) containing links to the web page.
8.	Alexa	Alexa Rank	Global Alexa rank of webpage
9.		Alexa Links number	Number of links to the web page
10.		Country Rank	Alexa Rank in the popular country.
11.	Majestic SEO	Citation Flow	Uses site link counts to the web page to see how influential the page is.
12.		Trust Flow	Trustworthiness of the page based on link to trustworthy neighbors.
13.		External Back Links	Total external back links to the web page
14.		Referred Domains	Total unique domains having links to the website
15.	SemRush	SemRush Rank	Domain rank by SemRush
16.		URL Links	Links to the mentioned web page
17.		Hostname Links	Links to the domain
18.	Ahrefs	URL Rating	Strength of web pages' back link profile and its chances of being ranked high in Google.
19.		Domain Rating	Strength of website's domain back link profile.
20.		Ahrefs Rank	Ranking based on size and quality back link profile
21.		Live & Fresh Index	List of live and dead links for the website

The dataset is divided into two equal sets and 500 influencer websites each having a spam score less than 5 and greater than 5 are taken as sample for conducting a statistical t-test to identify metrics that are significantly different in the two sets. Since, the range of values of each of the metrics is considerably varied; min-max normalization is used to standardize the data to a 0-1 range. Subsequently t-test is conducted and the metrics having a p-value less than 0.05 are

considered insignificant for further analysis. The final dataset for analysis thus comprises of the 12 significant attributes namely Domain Authority (DA), Page Authority (PA), Moz Rank (MR), Links In (LI), External Equity Links (ELL), Alexa Rank (AR), Citation Flow (CF), Trust Flow (TF), External Back Links (EBL), Referred Domains (RD), SemRush URL Links (UL) and SemRush Hostname Links (HL) for 2751 influencer websites. The p-values are illustrated in Table 5.

Table 5. Statistically Significant Metrics with p-value

Metric	P-value	Metric	P-value
Domain Authority	0.038	Citation Flow	7.23E-28
Page Authority	0.030	Trust Flow	0.0003
Moz Rank	1.03E-15	External Back Links	0.002
Links In	0.048	Referred Domains	3.94E-32
External Equity Links	1.25E-05	SemRush URL Links	7.12E-28
Alexa Rank	9.23E-15	SemRush Hostname Links	0.045

Further, bio inspired algorithms have been one of the most popular optimization techniques and mimic swarm behavior for optimization problems (Binitha & Sathya, 2012; Kar, 2016). Tang et al. (2012a) thus integrate a few popular bio inspired algorithms with K-means to avoid the local convergence. This study thus utilizes the integrated bio inspired wolf search algorithm for outlier detection.

2.2.1 Wolf Search Algorithm

The wolf search algorithm (WSA) is one such optimization approach that is said to overcome local optima by imitating the wolf preying behavior (Tang et al., 2012a, 2012b). In the current study, the number of clusters is identified as 2 for normal and outlier data points. The wolf population is initialized with visual distance and escape probability. The initial centroids are assigned for the two clusters. The fitness for the centroid in each wolf is calculated and the best solution is identified. The random preying behavior of the wolf is done by selecting a companion having the best solution within the visual distance. If the fitness of the companion is better than the self fitness of the wolf the companion is selected and is thus approached. After the prey is

hunted the wolf randomly selects a position beyond the visual range and the process is repeated from the new location. The centroids with the best fitness are considered as the final solution.

2.2.2 Bat Algorithm

Further, the results are compared with the integrated bat algorithm (BA) which uses the echolocation behavior of bats to find the prey and differentiate between different insects even in the dark (Yang, 2010b). The bat algorithm is one of the most popular algorithms used for several engineering, multi-objective and constrained optimization problems. For the integrated bat approach along with the two clusters, the bat population, frequency factor and loudness are initialized. The initial clusters are randomly assigned or the bat population. For each bat, the initial centroids are similarly identified. The fitness of the centroids is computed and the best solutions are identified. Further, the new solution is generated by adjusting the frequency and velocity. If the randomly generated solution is greater than the defined pulse rate, a new best solution is selected from the best solutions from each of the bats. The new solutions are accepted by adjusting pulse rate and loudness for subsequent iterations. The pulse rate is increased and the loudness is decreased for the next iteration.

Thus the bio-inspired algorithms help in identifying the best cluster centroids over iterations. The formulation of centroids is mainly iteratively guided by the search agents in the mentioned approaches. Since the dataset considered for this study requires only two clusters and has a total of 13 attributes for which the centroid values need to be computed. The $Centroid_{ij}$ is value of the centroid for i^{th} cluster and j^{th} attribute.

$$Centroid_{ij} = \sum_{k=1}^{SolSpace} weight_{ki} datapoint_{kj} \Big/ \sum_{k=1}^{SolSpace} weight_{ki} \tag{2.1}$$

The centroids largely depend on the weight that tells whether the data point belongs to the cluster or not. $weight_{ki} = 1, if\ datapoint_k \in cluster_i; else\ weight_{ki} = 0$. Once the best cluster centroids are identified for the two clusters of outliers and normal data points, a distance measure is subsequently used segregate the outliers.

The results for the two approaches used for the purpose show that the bat algorithm shows higher accuracy. Out of 2751 influencer websites, 1254 websites were identified as outliers based on their spam score. The BA correctly identified 1218 giving an accuracy of 97.12% while the WSA correctly identified 1203 with an accuracy of 95.93%. However, time taken to converge to the optimum solution is 22.61 seconds for BA while it is just 16.18 seconds for WSA.

CHAPTER 3

OUTLIER DETECTION IN UNSUPERVISED SCENARIO

This section of thesis is surrounding case scenarios surrounding outlier detection in an unsupervised scenario where the dataset does not have an output label for outliers. The algorithms proposed for this chapter include hybrid artificial bee colony and grey wolf optimization along with k-nearest neighbors. The sections considers social media datasets from Twitter for identifying outliers in the context of buzz and fake profiles.

3.1 Identifying Buzz in Social Media

The current era of digitization has completely revamped the way people engage with each other. Web 3.0 has not only enabled better engagement among individuals but has also opened new avenues for organizations to promote and propagate content across the globe. Organizations have realized that content popularity plays a vital role when it comes to engaging with a larger audience. However, not every piece of content gets the desired outreach. Content popularity is largely governed by metrics including the time the content is shared, the emotion, popularity and topic to which it pertains amongst others. Further, it becomes critical to analyze this large volume of content to understand the drivers for predicting what kind of content gains traction.

Content buzz in social media plays a vital role in the domain of social media marketing. The emergence of Web 3.0 has enabled marketers to creatively engage with huge consumer base by following internet marketing strategies on social media (Berthon et al. 2012). These web based social media applications have facilitated viral marketing and eWOM for information diffusion (Hausmann, 2012). Literature highlights frameworks that have been built around social media for online marketing (Leskovec et al. 2007). These frameworks enable web page ranking and brand popularity analysis (Aggrawal et al. 2017). The concept of virality simply extends from the diffusion of information. When this content reaches a larger audience at a faster pace, it creates a

"buzz" on the web.

There are several metrics that may have an impact on content virality like structure of the social network, behavior of users and the topics of discussion (Centola 2010; Weng et al 2013; Morales et al. 2014; Roelens et al. 2016; Tsugawa and Ohsaki 2017). Further, studies demonstrate virality frameworks that include concepts of spreadability, propagativity, integration and nexus among various platforms (Mills 2012; Aswani et al. 2017). The drivers of virality may be text, image, video or the trending meme (Guerini et al. 2011; Guerini et al. 2013; Deza and Parikh 2015; Ienco et al. 2010; Weng et al. 2012; Weng et al. 2014). None of the studies use heuristic/meta-heuristic approaches to address content buzz and segregate it.

The study models the 11 attributes (Table 6) to identify outliers in the form of "buzz" that are then validated using the output variable for measuring accuracy of the proposed approach. Since the values for each attribute belong to a different ranges, the dataset is normalized by scaling the values in the range of [0, 1] using a min-max normalization depending upon the correlation to the predicted output. The data is randomly sampled for training and testing the proposed model.

Table 6. Attributes used for Mining Twitter Buzz

Attribute	Category	Attribute Name	Description
A1	Network Level	Number of Created Discussions	Measure of the number of discussions created within a specific time (e.g. 1 hour of the tweet getting posted) on a tweet's topic.
A2		Author Increase	Number of new contributors who react to that tweet within a specific time (e.g. 1 hour of the tweet getting posted).
A3		Attention Paid	Measures the attention/number of views that the tweet gets on social media within a unit time. It is affected by the number of interacting contributors.
A4		Burstiness Level	Measure of the ratio of created discussions to the mean length of each discussion in terms of a unit of time.
A5	t S p e	Atomic	Measure of number of containers/topics of discussion

		Containers Tweet	surrounding the individual tweet.
A6		Attention Level Tweet	Measures attention that is paid to the tweet on its creation based on the topic of the tweet.
A7	Topic Level	Contribution Sparseness	Measure of contribution spread over discussions for the particular topic over a period of time.
A8		Author Interaction	Measure of the average number of authors that interact on the topic within discussions created by individual tweets.
A9		Number of Authors	Measure of the number of authors interacting within a specific time period (e.g. 1 hour of the tweet getting posted) on a specific topic.
A10		Mean Discussions Length in time	Measure of the mean discussion length in time that is incited by a topic from all the contributors of the topic.
A11		Average Number of Discussions	Measure of the average number of discussions that involve the topic within a specified time.

3.1.1 Artificial Bee Colony

Artificial bee colony optimization (ABC) provides a search mechanism based on population where food sources are searched by artificial bees (Karaboga and Basturk 2007a). The approach has been modified over the years for various application domains. It is known to give promising results in variety of domains (Karaboga and Basturk 2008). The approach is based on the intelligent behavior of bees. The colony of bees primarily comprises of three types of bees including onlookers, scouts and employer bees. The categorization is on the basis of how they choose the food source. ABC is a simple optimization approach that uses metrics like colony size, food sources, and foraging cycles as control variables. The goal of the bees is to discover food sources with highest nectar amount which is done by a search in the multidimensional space by updating their positions. The selection of the food source primarily depends on the type of bee. It is either by their own experience (employed bee) or by the experience of their mates (onlookers) who dance to show case the food sources with the highest amount of nectar. The

scouts, which is the last category of the bees however chooses the food sources by moving randomly without any experience. The new food source position is memorized only in the case when the nectar amount of the candidate solution is greater than the previous one.

The ABC is often used for optimization problems and a possible solution is represented by the position of a food source (Karaboga and Basturk 2007a; Karaboga and Basturk 2007b). The quality or the solution which is indicative of the fitness, is the nectar amount ($NecAmt$) of the food source. There is one employed bee for every food source in the hive. The employed bees modify the position in their memory depending upon the locally available visual information. This is done by testing the nectar amount which acts as the fitness of the food source under consideration. The modification in the position is done only if the nectar amount appears to be higher. The food source chosen by the onlooker bee is dependent on the value of probability corresponding to the food source Fp_i , computed using:

$$Fp_i = \frac{NecAmt_{\ i}}{\sum_{n=1}^{FdSrc} NecAmt_{\ i}} \tag{3.1}$$

$NecAmt_i$ depicts the fitness of the solution, indicative of the nectar amount of the source corresponding to the position i. The $FdSrc$ denotes the onlooker bees available in the bee hive. In this context, fitness represents the error while clustering the data points together and it becomes a minimization problem. The candidate solution (CFp_{ij}) is produced from the existing one in case the fitness is higher using:

$$CFp_{ij} = pos_{ij} + \tau_{ij}(pos_{ij} - pos_{kj}) \tag{3.2}$$

where $k \in [1,2 \ldots . \in n]$,n = number of employed bees, $j \in [1,2 \ldots . P]$, P being the number of optimization parameters, dimension of the solution and τ_{ij} is a random number ranging between [-1,1], having control over the neighborhood of pos_{ij}. The difference ($pos_{ij} - pos_{kj}$) reduces after every progressive iteration as the optimum solution is reached. Table 7 demonstrates the proposed k-nearest neighbor integrated artificial bee colony approach.

Table 7. Pseudo-code of k-nearest neighbor integrated Artificial Bee Colony Approach

Pseudo-code of k-nearest neighbor integrated artificial bee colony approach
Begin
Initialize the population, colony size, number of food sources and foraging cycles.
Initialize the initial cluster centers randomly
Repeat
For each set of data $k = 1 \ldots s$
/* 's' sets of employed bees */
do /*onlooker bees */
/*Finding the best food source using the experience of employed bees*/
Pass the new food source position (CFp_{ij}) computed to k-nearest neighbor fitness function to minimize F_n=d$(Ti, p) = \sqrt{\sum_{i=1}^{n}(Ti - p)^2}$, where Ti is the cluster centroid of ith cluster and p is the data point under consideration (the new food source) and get the output.
Assign the (data point) discussion to the cluster (normal or buzz) based on the distance
βi is the best instance, data point having minimum distance (dist) to the assigned cluster [min($\sum_{k=1}^{s} dist_k$)] for buzz. The βi has the dimensionality equal to number of attributes.
The βi is thus updated iteratively.
Move to the next set of data instances
End For
For each instance
Mark data point ($datap$) as outlier if it lies beyond outlier threshold computed using βi and standard deviation.
End

The data comprises of the final output value that is modeled using the 11 independent attributes. The final attribute is the number of active discussions denoting the popularity of the instance, the data distribution for the same is illustrated in Fig. 3. The "buzz" discussions are highlighted by the histogram in red. The break point is clearly depicted and corresponds to values greater than $\mu + 2\sigma$. The Twitter discussions are arranged in ascending order of buzz level to demonstrate the "buzz" beyond the identified threshold. Thus, any instance having the number of active discussions beyond $\mu + 2\sigma$ is considered as "buzz" in our study. The remaining instances are the

"non-buzz" data points in terms of activity and visibility. Fig.4 illustrates the outliers detected through the algorithm in terms of buzz and non-buzz.

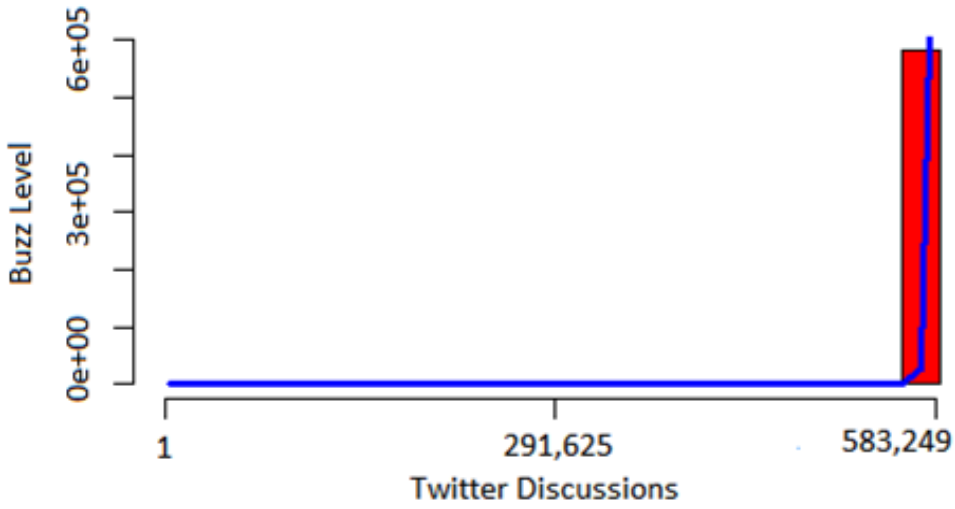


Fig. 3. Graphical distribution of buzz (based on activity incited by individual tweets)

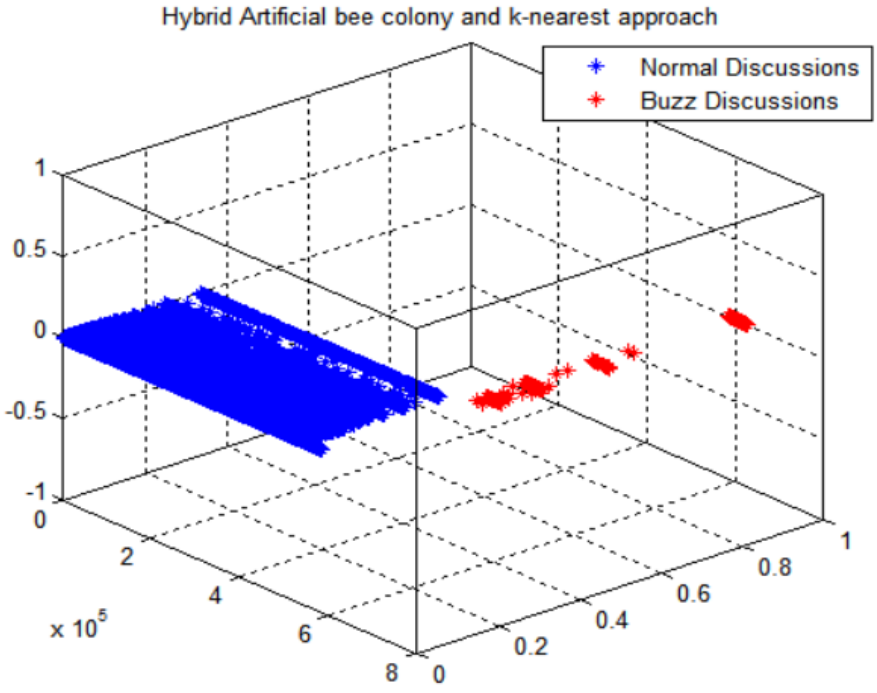


Fig. 4. Outlier plots for the Proposed ABC-kNN Approach

The comparative results with existing regression random forests approach is depicted in Table 8. It is evident that proposed approach outperforms the existing approach in terms of accuracy.

Table 8. Comparative Results for Buzz in Twitter

Approach	Algorithm	Accuracy
Proposed	ABC-kNN	97.8%
Existing (Kawala et al., 2013)	Regression Random Forests	94.2%

3.2 Detecting Fake Profiles on Social Media

A large number of social media platforms exist like Facebook, Digg, Twitter, LinkedIn, Youtube and Instagram to name a few and these have gained immense popularity in the last decade (Kaplan & Haenlein, 2010; Mui & Whoriskey, 2010). Due to ever increasing amount of information available on these networks, the users are forced to compete for attention and influence by relying on other people in their network to spread their message (Romero et al., 2011). Organizations and individuals are now trying to leverage these social media platforms to expand their content reach and try to boost the posted content to make it go viral. Therefore, for digital marketing, many organizations are resorting to strategies where they can increase the list of users that follow and share their information.

As a result a lot of fake profiles have been created on these social media platforms to boost follower count artificially and also to boost social votes and shares. A study reported that 44% of Facebook and 33% of Twitter followers have been identified to be fake (Taylor, 2012). Infact, fake Twitter followers has become a multimillion dollar business says New York Times (Perlroth, 2013). Individuals and organizations are paying for getting a huge follower list (Hockenson, 2012, Ritson, 2013). Literature highlights studies that consider followers for analyzing Twitter data making it essential to identify authentic versus fake Twitter followers. There have been studies for identifying fake profiles that discuss that fake profiles can be dangerous for society, politics and the country economy as they may try to propagate misleading information (Chu et al., 2012; Fire et al., 2014; Cresci et al., 2015). However the use of user personality dimensions for detecting fake profiles is still unexplored to the best of our knowledge.

For the purpose of this study, various aspects of social media, social sciences and bio inspired computing had to be taken into consideration. The study majorly focused on Twitter data and how personality dimensions can be used to identify fake Twitter profiles. The personality dimensions have been taken from the big five framework and the twitter statistics have been categorized into one of the five personality traits namely extroversion, conscientiousness, agreeableness, openness to experience and neuroticism (Aaker, 1997). The data collection for this study was done using the APIs in R. The 12 metrics that were finally achieved through the consensus from the Delphi Study comprised of HTML link count, unique words, statuses count, favorites count, @ mentions, follower count, friends count, hashtag frequency, emotion stability, polarity stability, unique hashtags and lexical diversity. The metrics are defined in Table 9. The dataset that is finally considered for the analysis comprises of 10,000 data points for a group of 10,000 unique Titter handles that have been actively propagating, sharing and liking viral content on Twitter.

Table 9. Twitter metrics Fake Profile Identification

S. No.	Metric Name	Metric Description
1.	Status Count (M9)	A count of statuses updated by the user.
	Unique Words (M6)	A metric that measures average number of unique words in each status.
	HTML Link Count (M5)	A metric that measures average number of HTML links in each status.
2.	Favorites Count (M10)	A count of number of tweets a user has liked.
	@ Mentions (M7)	A metric that counts @ mentions of other users in 50 recent tweets.
3.	Friends Count (M11)	A count of number of users the user follows.
	Hashtag Frequency (M3)	A metric that measures average number of hashtags in the last 50 tweets.
	Follower Count (M12)	A count of number of followers of the user.
4.	Emotion Stability (M1)	A metric that measures the deviation in emotional score for the given emotions-joy, sadness, surprise, disgust, fear and anger for a user in the last 50 tweets.
	Polarity Stability (M2)	A metric that measures the deviation in polarity score for

		both negative and positive polarity for a user in the last 50 tweets.
5.	Unique Hashtags (M4)	A metric that count the number of uniquely used hashtags in 50 recent tweets.
	Lexical Diversity (M8)	A metric that defines the fraction of unique words upon the total words used by a user in 50 recent tweets.

This study uses integrated approaches for detecting the outliers. The k-nearest neighbor approach has been used for detecting outliers and has proven to be simple yet effective (Yang and Huang, 2008). But they may fall in local optima and thus, bio inspired optimization techniques can be integrated with the basic nearest neighbor approach to give a global solution over specified iterations. This study uses grey wolf optimization (Mirjalili et al., 2014) and artificial bee colony (Karaboga and Basturk, 2007a; Karaboga and Basturk, 2007b; Karaboga and Basturk, 2008) approaches integrated with the k-nearest neighbors. The use of hybrid bio-inspired computing techniques in the domain of web 2.0 and social media including buzz identification, influencer selection and popularity prediction has shown promising results and thus this study explores the same for identification of fake profiles.

The artificial bee colony algorithm is known to produce promising results for solving single objective numerical problems. The algorithm works well for minimization and maximization problems and thus has been selected to for this study (Karaboga, 2005; Karaboga et al. 2014; Chakraborty and Kar, 2016). The grey wolf optimizer on the other hand, is known for its high exploration and exploitation function that assists it in outperforming other trainers for classification scenarios. Both of these have proven to be effective in numeric optimization problems.

3.2.1 Grey Wolf Optimization

The grey wolf optimization method mimics the hunting behavior of wolves, for identifying the most optimal solution (α), second best (β) and third best solution (δ) (Mirjalili et al., 2014; Aswani et al., 2016). The remaining solutions are referred to as omega (ω). The process of

hunting a prey (a workable solution in this case) is often guided by α wolves and the remaining follow. The process comprises of encircling the prey and subsequently hunting the prey to reach to a globally optimal solution over iterations, for the identified problem. The same can be mathematically modeled as:

$$\vec{X} = \left|\overrightarrow{V}.\ \vec{P}_{prey}\,(t) - \overrightarrow{P}(t)\right| \tag{3.3}$$

$$\vec{P}\,(t+1) = \vec{P}_{prey}\,(t) - \vec{U}\,.\vec{X} \tag{3.4}$$

The vector $\vec{X}$ is used to compute the distance between the position of the prey $\vec{P}_{prey}\,(t)$ in the current iteration t and the position of the wolf ($\overrightarrow{P}(t)$). Further, the position of the wolves is updated based on the computed vector $\vec{X}$ in the subsequent iteration$(t+1)$. $\vec{U}$ and $\overrightarrow{V}$ are coefficient vectors and are defined as:

$$\vec{U} = 2\,\overrightarrow{u}\,.\vec{d}_1 - \vec{u} \tag{3.5}$$

$$\vec{V} = 2.\vec{d}_2 \tag{3.6}$$

The coefficient vectors are computed using vector $\vec{u}$ which decreases from 2 to 0 in a linear fashion and the two random vectors $\vec{d}_1$ and $\vec{d}_2$ which vary in the range [0, 1].

Once the prey is encircled by the grey wolves, the distance vector is computed for each of the wolf using the position of α, β and δ wolves as the location of the prey (the optimum solution) is not definite in the abstract search space. The equations (3.7), (3.8) and (3.9) are descriptive of the step size while the equations (3.10), (3.11) and (3.12) are used to represent final positions of the remaining wolves.

$$\vec{X}_\alpha = \left|\vec{U}_1.\ \vec{P}_\alpha - \vec{P}\right| \tag{3.7}$$

$$\vec{X}_\beta = \left|\vec{U}_2.\ \vec{P}_\beta - \vec{P}\right| \tag{3.8}$$

$$\vec{X}_\delta = \left|\vec{U}_3.\ \vec{P}_\delta - \vec{P}\right| \tag{3.9}$$

$$\vec{P}_1 = \vec{P}_\alpha - \vec{V}_{1.}\,(\vec{X}_\alpha) \tag{3.10}$$

$$\vec{P}_2 = \vec{P}_\beta - \vec{V}_{2.}\,(\vec{X}_\beta) \tag{3.11}$$

$$\vec{P}_3 = \vec{P}_\delta - \vec{V}_{3.}(\vec{X}_\delta) \quad (3.12)$$

where, every dot product $\vec{A}.\vec{B} = a_1 b_1 + a_2 b_2 \ldots\ldots + a_n b_n$ (3.13)

It is thus assumed that the best, second best and third best wolves (candidate solutions) have greater knowledge about the potential prey position. Thus these three groups of wolves oblige the remaining ω wolves for updation of their positions according to the best search wolves in the next iteration$(t+1)$.

$$\vec{P}(t+1) = \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \quad (3.14)$$

3.2.2 Artificial Bee Colony

The artificial bee colony approach on the other hand works on the behavior of honey bees and how they identify food sources (optimal solution) (Karaboga and Basturk, 2007a). The solution quality is governed by the availability of nectar amount (N) on an identified food source. The approach works on three different types of bees including employed bees, onlooker bees and scouts. The employed bees are the ones that guide the others by dancing on the floor to inform the location of the best food source. The onlooker bees take the decision based on the experience of the employed bees and their guidance. The food source is selected based on the probability value associated with it which is computed:

$$Prob_i = \frac{N_i}{\sum_{n=1}^{Fd} N_i} \quad (3.15)$$

The N_i represents the fitness value of the candidate solution indicating the amount of nectar of the source corresponding to the current position i. The value Fd corresponds to the count of onlooker bees in the hive which is equal to the food sources. In the context of the current study the fitness value of the solution is representative of the error in grouping the profiles into being fake or authentic and thus becomes a minimization problem. The candidate solution (FdS_{ij}) is obtained using:

$$FdS_{ij} = PrevFdS_{ij} + r_{ij}(PrevFdS_{ij} - PrevFdS_{kj}) \quad (3.16)$$

where $\in [1,2 \ldots . \in n]$, $j \in [1,2 \ldots . P]$, P is solution dimension and r_{ij} is rand[-1,1] controlling neighborhood of $PrevFdS_{ij}$.

This study uses a hybrid k-nearest neighbor integrated grey wolf optimization approach (GWO-kNN) to segregate fake Twitter profiles. The GWO-kNN approach for outlier detection in the form of fake profiles is used in this study. The results are compared with a similar k-nearest neighbor and bio inspired integrated approach (ABC-kNN) (Aswani et al. 2017b). Table 10 illustrates the pseudo-code for the integrated approaches GWO-kNN and ABC-kNN, which was necessary for meeting the objectives of the current study.

GWO-kNN	**ABC-kNN**

Begin
Step 1: Initialization. Start iteration counter with t=1
Initialize population of grey wolves randomly;
Initialize vectors $\vec{u}, \vec{U}$ and $\vec{V}$
Initialize search agents $\vec{P}_{\alpha}$, $\vec{P}_{\beta}$ and $\vec{P}_{\delta}$
Determine number of groups, K.
Assign K clusters for each wolf.
Assign K objects as initial centroids for each group.
Step 2: Exploration. Compute fitness of centroids for each group
While t < MaxGeneration do
Sort the population according to their fitness.
Update position of each search agent
Update vectors $\vec{u}, \vec{U}$ and $\vec{V}$ (Equation 3 and 4)
Evaluate fitness of each search agent
Step 3: Centroid Update. Sort the population and find current best solution.
	Begin
	Step 1: Initialization. Start iteration counter with t=1
	Initialize population of bees randomly;
	Initialize colony size and food sources.
	Determine number of groups, K.
	Assign K clusters for each wolf.
	Assign K objects as initial centroids for each group.
	Step 2: Exploration. Compute fitness of centroids for each group
	While t < MaxGeneration do
	Sort the population according to their fitness.
	Produce updated solutions (FdS_{ij}) for the employed bees
	Calculate probability $Prob_i$ for onlookers
	Evaluate fitness of the obtained solutions
	Identify abandoned scout solutions for replacement
	Step 3: Centroid Update. Sort the population and

t=t+1. Reassign the clusters. Output best cluster configuration based on best fitness. **End while** **End**	find current best solution. t=t+1. Reassign the clusters. Output best cluster configuration based on best fitness. **End while** **End**

Table 10. Pseudo-code of GWO-kNN and ABC-kNN

Independent Samples Test							
	Levene's Test for Equality of Variances		t-test for Equality of Means				
	F	Sig.	t	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
						Lower	Upper
M1	1.615	.211	13.873	.68465	.04935	.58474	.78456
M2	7.553	.009	21.553	.80525	.03736	.72962	.88088
M3	6.858	.013	23.874	.80395	.03367	.73578	.87212
M4	9.675	.004	50.837	.77105	.01517	.74035	.80175
M5	13.373	.001	28.235	.59370	.02103	.55113	.63627
M6	.114	.737	13.656	.44005	.03222	.37482	.50528
M7	1.067	.308	17.916	.53435	.02982	.47397	.59473
M8	10.333	.003	10.599	.47450	.04477	.38387	.56513
M9	.445	.509	14.453	.38625	.02672	.33215	.44035
M10	14.903	.000	22.963	.47130	.02052	.42975	.51285
M11	.349	.558	8.201	.40365	.04922	.30401	.50329
M12	.032	.859	7.933	.41495	.05231	.30906	.52084

Fig. 5. Results for Statistical Validation Test

The Levene's test for equality of variances for the dataset containing 20 cluster centers in each group, the fake profiles (F) and the authentic profiles (A) identifies some of the metrics as insignificant.

The resulting p-value for the metrics Emotion Stability (M1), Unique Words (M6), @ Mentions (M7), Status Count (M9), Friends Count (M11) and Followers Count (M12) show insignificance. While, the two-tailed t-test for equal means shows the significance of all the metrics. This clearly shows that although the dispersion of cluster centers across the two groups of twitter users is same, the cluster centers are actually different data points as the means of the cluster centers show significant difference across all the 12 metrics. The results of the analysis are compared with the existing fake profile detection approaches and the parent approaches. Table 11 depicts the results with GWO-kNN outperforming ABC-kNN in terms of convergence which proved to be a disadvantage as it leads to premature convergence as discussed in literature and reduces the accuracy in return.

Table 11. Comparative Analysis of Fake Profile Detection

Approach	**Algorithm**	**Avg.Time (seconds)**	**Convergence Iterations**	**Accuracy %**
Existing (Galán-García et al., 2016)	SMO-Poly Kernel	N/A	N/A	68.47
	J48	N/A	N/A	65.81
	SMO-Normalized Poly Kernal	N/A	N/A	65.29
	Random Forest	N/A	N/A	59.79
	kNN k=10	N/A	N/A	59.7
	kNN k=3	N/A	N/A	59.39
	kNN k=5	N/A	N/A	33.91

	Baive Bayes	N/A	N/A	61.06
Parent (Mirjalili et al., 2014; Karaboga & Basturk, 2007; Cunningham & Delany, 2007)	GWO	87.56	**2**	**80.17**
	ABC	96.78	4	83.28
	kNN	120.23	4	**80.23**
Proposed	ABC-kNN	98.83	4	**87.09**
	GWO-kNN	89.65	**2**	83.04

CHAPTER 4

INTEGRATING CHAOS FOR OUTLIER DETECTION

This sections attempts to improvise on the convergence speed by exploring a better search space while optimizing through the uses of chaos theory. In almost all meta heuristic algorithms with stochastic components, randomness is achieved by using some probability distributions, often uniform or Gaussian. In principle, it can be advantageous to replace such randomness by chaotic maps because chaos can have very similar properties of randomness with better statistical and dynamical properties. Such dynamical mixing is important to ensure that solutions generated by the algorithm can be diverse enough to potentially reach every mode in the multimodal objective landscape. The Chaotic Optimization Algorithms (COA) use chaotic variables in random-based optimization and is known to carry the overall search at higher speeds primarily because of the non-repetition of chaos (dos Santos Coelho and Mariani 2008). Due to the mixing properties of chaos, algorithms can potentially carry out iterative search steps at higher speeds than standard stochastic search with standard probability distributions. The chapter introduces chaotic firefly and cuckoo search approaches for segregating popular online content and identifying spam in search engines.

4.1 Segregating Popular Online Content

The data set used for the analysis comprises of 39,797 articles from a popular content publishing website called Mashable. Each article is expressed by a set of 58 predictive parameters that includes descriptive metrics pertaining to links, words, digital media (images and videos) and publishing time. It further comprises of metrics surrounding sentiment and polarity of keywords, topics and several metrics that are extracted using advanced text mining techniques. The data is publicly available at the UCI repository (http://archive.ics.uci.edu/ml/). The articles focus on primarily six domains including “Tech”, “Entertainment”, “Social Media”, “Lifestyle”, “World” and “Business”. Since the values for the metrics belong to varied ranges, the data was normalized to 0-1 using a min max normalization approach. Further, based on the shares threshold the data is

divided into popular and not popular and a statistical t-test is conducted to identify significant metrics.

The Table 12 describes the significant metrics used for the purpose of analysis in the current study. It comprises of 19 independent metrics and one outcome metric which is used to predict content popularity. The study considers a 95% confidence interval and the metrics with a p-value less than 0.05 have been finally considered for analysis. The metrics are used to model and cluster the articles into popular vs. not popular. Further, the total shares metric is used for the purpose of validation of the proposed approach.

Table 12. Description of Metrics for Online Popularity

	Metric	**Description**
M1	Link count	The count of links embedded in the content
M2	Mashable link count	The number of links referring to Mashable articles
M3	Image count	Count of embedded images
M4	Video count	Count of embedded videos
M5	Keyword Count	Keyword count in the article
M6	Mashable article share	Number of times the article is shared in Mashable
M7	Closeness to topic 1	Closeness of article to “Tech” as obtained using topic modeling
M8	Closeness to topic 2	Closeness of article to “Entertainment” as obtained using topic modeling
M9	Closeness to topic 3	Closeness of article to “Social Media” as obtained using topic modeling
M10	Closeness to topic 4	Closeness of article to “Lifestyle” as obtained using topic modeling
M11	Closeness to topic 5	Closeness of article to “Business” as obtained using topic modeling
M12	Content subjectivity	Subjectivity of the article text
M13	Content sentiment	Sentiment of the article text

	Metric	Description
M14	Avg. positive polarity	Average positive polarity of the article
M15	Avg. negative polarity	Average negative polarity of the article
M16	Title subjectivity	Subjectivity of the article title
M17	Title polarity	Polarity of the article title
M18	Absolute subjectivity	Level of absolute subjectivity
M19	Absolute polarity	Level of absolute polarity
M20	Total shares	Target output variable for total number of shares

4.1.1 Chaotic Cuckoo Search Algorithm

The cuckoo search (CS) algorithm mimics the brooding behavior of cuckoos. The movement of cuckoos towards nests is done via Levy flights. The CS algorithm is primarily a population based search algorithm for finding a globally optimal solution. The cuckoos lay their eggs in the nest of the other birds referred to as host birds. Further, cuckoos also imitate the patterns of host birds to avoid abandoning their eggs by them. The cuckoo eggs in the nests are representative of a potential solution (Yang and Deb, 2009). A new solution is obtained by means of Levy Flights. The new solution (pos_i^{t+1}) is obtained using the previous candidate solution (pos_i^t) in combination with Levy ($Levy(\delta)$).

$$pos_i^{t+1} = pos_i^t + Step\ Size \oplus Levy(\delta) \quad (4.1)$$

where, s > 0

and $Levy \sim u = v^{-\delta}$ (4.2)

The step size in this case is updated using the chaotic maps. The current study considers a set of **twelve chaotic maps** for the analysis (Wang et al., 2016) including circle, intermittency, Gaussian, tent, sinusoidal, chebyshev, piecewise, logistic, sine, liebovitch, singer and iterative map. The initial step size is computed using the Mantegna's algorithm for a stable Levy flight (Mantegna & Stanley, 1994)

$$Step\ Size = \frac{u}{|v|^{1/\delta}} \tag{4.3}$$

The values $u \sim NormDist(0, {\sigma_u}^2)$ and $v \sim NormDist(0, {\sigma_v}^2)$, where $\sigma_u = \left[\frac{Gamma\ (1+\delta)\ sin\left(\frac{\pi\delta}{2}\right)}{Gamma\left[\frac{(1+\delta)}{2}\right]\delta 2^{\frac{\delta-1}{2}}}\right]^{\frac{1}{\delta}}$

and $\sigma_v = 1$, $\delta = 3/2$ describe the Levy flight.

The chaotic cuckoo search approach is integrated with k-means for clustering (Tang et al., 2012a). The proposed approach for outlier detection is illustrated in Fig. 6. The comparison among chaotic map variants id depicted in terms of execution time and convergence speeds (Fig.7 and Fig.8 respectively)

Begin
Step 1: Initialization. *Set the iteration counter t = 1.*
Initialize the population of host nest P randomly;
Set discovery rate (d) and initial value of the chaotic map randomly.
Determine number of clusters, K.
Assign K clusters for each host nest P
Assign K objects as initial centroids for each nest.
Step 2: Exploration. *Calculate fitness of centroid for each cluster*
While *t < Max* ***do***
Sort the population according to their fitness.
Update the step size using chaotic maps
Select a cuckoo i randomly
Replace the corresponding solution by Lévy flight (Equation 1-4).
Evaluate its fitness Fit_i.
Identify a nest j randomly.
if ($Fit_i < Fit_j$)
Replace j by the newly obtained solution.
end if
Abandon fraction (d) poor nests and build new nests.
Step 3: Centroid Update. *Sort the population and find the current best.*
t = t+1.
Reassign the cluster centers
Output best cluster configuration based on best fitness value.
end while
End.

Fig 6. Pseudo-code of k-means integrated Chaotic Cuckoo Search

S. No.	Running time for Chaotic Cuckoo Search variants	
	Variant	*Time elapsed (seconds)*
1.	Circle Map	282.6
2.	Intermittency Map	276.9
3.	Gaussian Map	403.9
4.	Tent Map	242.3
5.	Sinusoidal Map	217.2
6.	Chebyshev Map	280.7
7.	Piecewise Map	285.9
8.	Logistic Map	258.8
9.	Liebovitch Map	245.1
10.	Singer Map	274.1
11.	Iterative Map	320.7
12.	Sine Map	254.9
13.	CS without Chaos	239.1

Fig 7. Execution Time for Chaotic Variants

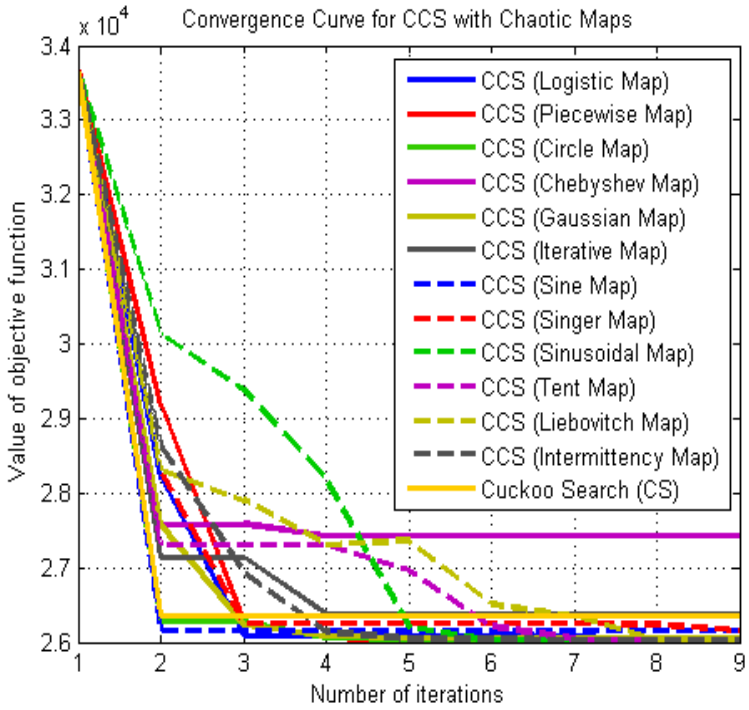


Fig 8. Comparison for Convergence Plots for Chaotic Variants

Further, the proposed hybrid bio-inspired approaches are compared with existing traditional machine learning approaches in literature in terms of accuracy, precision, recall and the F-measure. The chaotic variants outperform the same along with the non-chaotic cuckoo search approach. the results are highlighted in Table 13.

Table 13. Comparative Analysis for Online Content Popularity

Approach	Model	Accuracy	Precision	Recall	F-Measure
Existing (Fernandes et al., 2105)	Random Forest (RF)	0.67	0.67	0.71	0.69
	Adaptive Boosting (AdaBoost)	0.66	0.68	0.67	0.67
	Support Vector Machine (SVM)	0.66	0.67	0.68	0.68
	K-Nearest Neighbors (KNN)	0.62	0.66	0.55	0.60
	Naïve Bayes (NB)	0.62	0.68	0.49	0.57
Proposed (Yang & Deb, 2009)	Cuckoo Search (w/o Chaos) [10]	0.83	0.83	0.77	0.79
	Cuckoo Search (Singer Map)	**0.86**	**0.84**	0.79	0.81
	Cuckoo Search (Sine Map)	0.84	0.83	**0.82**	**0.82**

4.2 Identifying Spam Websites in Search Engines

Literature highlights the important factors for online marketing and their relevance in making the content popular among the users (Kiang et al., 2000). Further, a lot of traffic to these organization websites comes from organic search queries in search engines like Google (Brin and Page, 1998), Bing and Yahoo to name a few. Search engines prove to be a great tool in quest to locate online information by people. People generally tend to use the top results from their search query, making the rank of web pages of vital importance for the companies. About 62% of users tend to use the results from the first page of their search query results and only about less than 10% users go beyond the third page for the desired content (Malaga, 2008).

As a result, organizations have started adopting brand positioning strategies using search engine marketing (Dou et al., 2010). Organizations tend to hire service providers for improving their rank in organic search results through content and link creation (Aswani et al., 2018) . Several low cost techniques like article spinning, link farming and keyword stuffing are popular for recreating, manipulating and building content for link building are usually used by service providers. Such techniques are often not beneficial for business and Google has introduced several changes and algorithms to weed out such spam websites in the past. Mechanisms that were intended to specifically penalize and ban sites that used these techniques, Google Florida, Google Panda and Google Hummingbird were introduced to weed out such spam pages (Slegg, 2016).

The methodology comprises of several steps including the initial cluster initialization, followed by optimization of initial cluster centers using the bio inspired approaches and subsequent clustering of the training data (60%) for the two cases into two clusters of outlier and authentic websites for SEM. Subsequently he remaining 40% of the data is classified using k-nearest neighbor approach after manual validation of the two clusters. The sub sections highlight the cluster initialization, bio inspired clustering and post data classification.

The K-Means clustering approach computes the initial cluster centers for the two clusters in this study by considering the significance in terms of belongingness to the cluster along with each website data point (webdp) and is given by:

$$significance_{i,K} = \begin{cases} 1, & webdp_i \in cluster_j \\ 0, & webdp_i \notin cluster_j \end{cases} \quad (4.4)$$

$$ClusterCenter_{K,M} = \frac{\sum_{i=1}^{SolSpace} significance_{i,K}\, webdp_{i,M}}{\sum_{i=1}^{SolSpace} significance_{i,K}}, \quad (4.5)$$

where K=1…k, k being the number of clusters, SolSpace refers to the solution space and M=1…k*m, m is the number of metrics under consideration. The subsequent distance between the cluster centers is given by:

$$F(ClusterCenter) = \sum_{K=1}^{k} \sum_{i=1}^{SolSpace} significance_{i,K} \sum_{M=1}^{k*m} \left(webdp_{i,M} - ClusterCenter_{K,M}\right)^2 \quad (4.6)$$

The initial cluster centroids are then updated over iterations by means of proposed chaotic firefly algorithm and with a minimization objective function depicted by:

$$F = \sum_{K=1}^{k} \sum_{i=1}^{Population} \left|\left|webdp_{i,K} - ClusterCenter_K\right|\right|^2 \quad (4.7)$$

The cluster initialization and subsequent clustering using the integrated approach is done for 60% data for the two cases, 1009 website data points for Case-1 and 642 data points for Case-2.

4.2.1 Chaotic Firefly Algorithm

The cluster centers achieved in the initialization phase are then updated iteratively to produce a globally optimum solution by means of bio inspired computing approach. The firefly algorithm (FA) mimics the behavior of fireflies that move towards each other based on their brightness (Yang 2010a).

We further use chaotic maps to speed up the search for best solution in the environment. The chaos theory can be used in two ways in the firefly algorithm (Gandomi et al. 2013) which can be used to tune the absorption coefficient μ and the attractiveness coefficient A. It is observed that Sinusoidal and Gauss map are known to give best results for the two mentioned scenarios respectively. The sinusoidal map for tuning absorption coefficient μ can be represented as:

$$x_{i+1} = \tau x_i^2 sin(\pi x_i), \quad (4.8)$$

$\tau = 2.3,\ x_o = 0.7$ and i represents the i^{th} chaotic map.

While, the Gauss map for tuning the attractiveness coefficient A is:

$$x_{i+1} = \begin{Bmatrix} 0 & x_i = 0 \\ 1/_{imod(1)} & x_i \neq 0 \end{Bmatrix}, \text{ where } {}^{1}/_{x_i mod(1)} = {}^{1}/_{x_i} - \left[{}^{1}/_{x_i}\right] \quad (4.9)$$

Once the best solutions are identified using the bio inspired approaches, it is considered as the cluster centroids for the K-Means integration. Further, k-nearest neighbors approach is used to mine the outliers. Euclidean distance is used as the measure to compute the distance between each websites corresponding attributes and the best solution. The proposed chaotic firefly approach is compared with bat algorithm which is known to converge to an optimum solution the fastest when integrated with K-Means for identifying cluster centers and cuckoo search algorithm that is known to give the best results as per existing literature (Tang et al., 2012a). Further, there are existing studies surrounding cuckoo search algorithm for website selection (Kar, 2014) which has been used for benchmarking in this study. A comparative analysis of the three approaches thus used for detecting website outliers along with the parameter values taken are illustrated in Table 14.

Table 14. Comparative Analysis of the Three Bio-inspired Computing Approaches

Chaotic Firefly Algorithm (FA)	**Bat Algorithm (BA)**	**Cuckoo Search (CS) Algorithm**

Begin
Generate initial population for fireflies, $N_i, i = 1 \ldots . n$
Light intensity ρ_i for each N_i determined by f(N)
Initialize light absorption coefficient μ
For all fireflies in the population,
If ($\rho_i < \rho_j$)
Move i firefly towards j;
$dist_{ij} = \left\|pos_i - pos_j\right\|$ is the
	Begin
	Generate initial population N_i and velocity, ω_i.
	Pulse frequency Pf_i at N_i is $Pf_i \in [0, Pf_{max}]$,
	Define Pulse Rate (P)
	Loudness (L)
	F: Random flight with velocity ω_i at position pos_i at time interval t
	$\omega_i^t = \omega_i^{t-1} + \left(pos_i^{t-1} - posbest * fi\right.$
		Begin
		Generate initial population of 'n' host nests, $N_i, i = 1 \ldots . n$
		Get a cuckoo i randomly by Levy Flights L
		Select a nest j among nests n
		Fitness of cuckoo i: Fi
		If (Fi > Fi)
		Replace j by new solution
		Abandon fraction of nests D_{rate} (worst nests) and build more solution pos_i at time t is

Euclidean distance between the two fireflies at positions $pos_i \; and \; pos_j$ The new position of the firefly, $pos_i += A\left(pos_j - pos_i\right) + \omega\beta_i$, that depends on β_i random variable vector and ω is the randomization coefficient. Attractiveness (A) can be varied $A = A_0\, e^{-\mu dist^2}$ which depends on brightness ρ: $\rho = \rho_0\, e^{-\mu dist}$ Keep the best solutions Rank fireflies and find current best solution.	and $pos_i^t = pos_i^{t-1} + \omega_i^t$ If (F > P) Select a solution among current best solutions (pos_{best}) If the (F < L) Accept new solution, reduce loudness $L_i^t = \tau L_i^{t-1}, \quad \tau \in (0,1)$ and increase pulse rate $P_i^t = P_i^{t=0}[1 - e^{-\phi t}]$, $\phi \; is \; constant > 0$. Keep the best solutions Rank bats and find current best solution.	the new solution by Levy L. $pos_i^{t+1} = pos_i^t + step \oplus L(\delta)$, step > 0, L ~$x = y^{-\delta}$ and $step = \frac{x}{\lvert y\rvert^{1/\delta}}$ x~ $NormDist(0, \sigma_x{}^2)$ and y~ $NormDist(0, \sigma_y{}^2)$ where $\sigma_x = \left[\frac{Gamma\ (1+\mu)\sin\left(\frac{\pi\mu}{2}\right)}{Gamma\left[\frac{(1+\mu)}{2}\right]\mu 2^{\frac{\mu-1}{2}}}\right]^{\frac{1}{\mu}}$ and $\sigma_y = 1$, where $\mu = 3/2$ (Mantegna, 1994) Keep the best solutions Rank the nests and find the current best solution.

After the optimum cluster centers are obtained the K-Means clustering is done to achieve the two cluster sets for authentic and outlier websites. Manual evaluation is used for validating these outlier websites. Subsequently the remaining 40% website data points are used for classification. The results are illustrated in Table 15 with a comparison in terms of accuracy measures and convergence iterations.

Table 15. Comparison in terms of Accuracy and Convergence

Approach	**Algorithm**	**Accuracy**	**Precision**	**Recall**	**F-measure**	**Convergence Iterations**
Proposed	Chaotic FA (Tuning Absorption)	**0.98**	0.94	**0.99**	**0.96**	**3**

	Chaotic FA (Tuning Attractiveness)	0.97	0.94	0.95	0.94	**3**
Existing	Firefly Algorithm (FA) [12]	0.96	0.93	0.94	0.93	7
	Cuckoo Search (CS) [10]	0.97	0.95	0.96	0.95	8
	Bat Algorithm (BA) [13]	**0.98**	**0.95**	0.97	0.95	4

CHAPTER 5

CONCLUSION & FUTURE SCOPE

In the current scenario where there is plethora of information available on the web and people are in a constant quest to know more, the information visibility becomes of critical importance. The Web 3.0 thus plays a key role in providing the content on the go. The study primarily targets social media and web analytics data to mine outliers. The use of meta-heuristics specifically bio inspired computing techniques is also a novel contribution of this work.

The work integrates traditional k-Means and k-nearest neighbors approaches with bio inspired algorithms including firefly, cuckoo search, bat, grey wolf optimizer, artificial bee colony and wolf search algorithms for detecting outliers. The hybrid approaches are known to avoid local optimum. The work also uses chaos theory and levy flight for better search space and faster results.

Use of Chaotic Optimization Theory

- In almost all meta heuristic algorithms with stochastic components, randomness is achieved by using some probability distributions, often uniform or Gaussian.
- In principle, it can be advantageous to replace such randomness by chaotic maps because chaos can have very similar properties of randomness with better statistical and dynamical properties.
- Such dynamical mixing is important to ensure that solutions generated by the algorithm can be diverse enough to potentially reach every mode in the multimodal objective landscape.
- The Chaotic Optimization Algorithms (COA)use chaotic variables in random-based optimization and is known to carry the overall search at higher speeds primarily because of the non-repetition of chaos (dos Santos Coelho and Mariani 2008).
- Due to the mixing properties of chaos, algorithms can potentially carry out iterative search steps at higher speeds than standard stochastic search with standard probability distributions.

Use of Levy Flights

- The Levy flights are known to maximize the search for resources in uncertain environments (Yang 2010a).
- As Levy flights have infinite mean and variance, the meta-heuristic approaches can explore the search space more efficiently than algorithms by standard Gaussian process.
- This advantage, combined with both local and search capabilities and guaranteed global convergence

When it comes to optimization algorithms, we cannot generalize on which approach is better. The No Free Lunch (NFL) theorem clearly states that an algorithm can perform extremely well on a dataset and the same algorithm may give abysmal results on the other one. The dataset domain and data distribution play an important role in the performance on an approach.

Future work is planned to use fuzzy clustering approaches along with the proposed approaches to highlight the overlapping clusters. The future work may also consider integrating the proposed approaches with big data platforms by using the Map Reduce Framework to further reduce the time required for executing the approaches.

REFERENCES

Aaker, J. L. (1997). Dimensions of brand personality. Journal of marketing research, 347-356.

Acuna, E., & Rodriguez, C. (2004). A meta analysis study of outlier detection methods in classification. Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez.

Aggrawal, N., Ahluwalia, A., Khurana, P., & Arora, A. (2017). Brand analysis framework for online marketing: ranking web pages and analyzing popularity of brands on social media. Social Network Analysis and Mining, 7(1), 21.

Alam, S., Dobbie, G., Riddle, P., & Naeem, M. A. (2010, July). A swarm intelligence based clustering approach for outlier detection. In Evolutionary Computation (CEC), 2010 IEEE Congress on (pp. 1-7). IEEE.

Aswani, R., Kar, A. K., Aggarwal, S., & Ilavarsan, P. V. (2017, November). Exploring Content Virality in Facebook: A Semantic Based Approach. In Conference on e-Business, e-Services and e-Society (pp. 209-220). Springer, Cham.

Aswani, R., Kar, A. K., Ilavarasan, P. V., & Dwivedi, Y. K. (2018). Search engine marketing is not all gold: insights from Twitter and SEOClerks. International Journal of Information Management, 38(1), 107-116.

Berthon, P. R., Pitt, L. F., Plangger, K., & Shapiro, D. (2012). Marketing meets Web 2.0, social media, and creative consumers: Implications for international marketing strategy. Business horizons, 55(3), 261-271.

Binitha, S., & Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. International Journal of Soft Computing and Engineering, 2(2), 137-151.

Bolton, R. J., & Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. Credit Scoring and Credit Control VII, 235-255.

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18), 3825-3833.

Centola, D. (2010). The spread of behavior in an online social network experiment. science, 329(5996), 1194-1197.

Chakraborty, A., & Kar, A. K. (2016). A Review of Bio-Inspired Computing Methods and Potential Applications. In Proceedings of the International Conference on Signal, Networks, Computing, and Systems (pp. 155-161). Springer India.

Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. In Nature-Inspired Computing and Optimization (pp. 475-494). Springer International Publishing.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 15.

Chawla, S., & Gionis, A. (2013, May). k-means–: A unified approach to clustering and outlier detection. In Proceedings of the 2013 SIAM International Conference on Data Mining (pp. 189-197). Society for Industrial and Applied Mathematics.

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg?. IEEE Transactions on Dependable and Secure Computing, 9(6), 811-824

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2015). Fame for sale: efficient detection of fake Twitter followers. Decision Support Systems, 80, 56-71.

Cunningham, P., & Delany, S. J. (2007). k-Nearest neighbour classifiers. Multiple Classifier Systems, 34, 1-17.

Deza, A., & Parikh, D. (2015). Understanding image virality. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1818-1826).

dos Santos Coelho, L., & Mariani, V. C. (2008). Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Systems with Applications, 34(3), 1905-1913.

Dou, W., Lim, K. H., Su, C., Zhou, N., & Cui, N. (2010). Brand positioning strategy using search engine marketing. Mis Quarterly, 261-279.

Fernandes, K., Vinagre, P., & Cortez, P. (2015, September). A proactive intelligent decision support system for predicting the popularity of online news. In Portuguese Conference on Artificial Intelligence (pp. 535-546). Springer, Cham.

Fire, M., Kagan, D., Elyashar, A., & Elovici, Y. (2014). Friend or foe? Fake profile identification in online social networks. Social Network Analysis and Mining, 4(1), 1-23.

Fister, I., Yang, X. S., & Brest, J. (2013). A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, 13, 34-46.

Forestiero, A. (2017). Bio-inspired algorithm for outliers detection. Multimedia Tools and Applications, 1-19.

Galán-García, P., Puerta, J. G. D. L., Gómez, C. L., Santos, I., & Bringas, P. G. (2016). Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. Logic Journal of the IGPL, 24(1), 42-53.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. Communications in Nonlinear Science and Numerical Simulation, 18(1), 89-98.

Garrigos, F. (2009). Interrelationships between professional virtual communities and social networks, and the importance of virtual communities in creating and sharing knowledge. In Connectivity and Knowledge Management in Virtual Organizations: Networking and Developing Interactive Communications (pp. 1-22). IGI Global.

Garrigos-Simon, F. J., Lapiedra Alcami, R., & Barbera Ribera, T. (2012). Social networks and Web 3.0: their impact on the management and marketing of organizations. Management Decision, 50(10), 1880-1890.

Gebremeskel, G. B., Yi, C., He, Z., & Haile, D. (2016). Combined data mining techniques based patient data outlier detection for healthcare safety. International Journal of Intelligent Computing and Cybernetics, 9(1), 42-68.

Guerini, M., Staiano, J., & Albanese, D. (2013, September). Exploring image virality in google plus. In Social Computing (SocialCom), 2013 International Conference on (pp. 671-678). IEEE.

Guerini, M., Strapparava, C., & Özbal, G. (2011, July). Exploring Text Virality in Social Networks. In ICWSM.

Hausmann, A. (2012). Creating 'buzz': opportunities and limitations of social media for arts institutions and their viral marketing. International Journal of Nonprofit and Voluntary Sector Marketing, 17(3), 173-182.

Hendler, J. (2009). Web 3.0 Emerging. Computer, 42(1).

Hockenson, L. (2012). Fake Twitter followers: An easy game, but not worth the risk.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial intelligence review, 22(2), 85-126.

Ienco, D., Bonchi, F., & Castillo, C. (2010, December). The meme ranking problem: Maximizing microblogging virality. In Data Mining Workshops (ICDMW), 2010 IEEE International Conference on (pp. 328-335). IEEE.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. Business horizons, 53(1), 59-68.

Kar, A. K. (2014). A Decision support system for website selection for internet based advertising and promotions. In Emerging Trends in Computing and Communication (pp. 453-457). Springer, New Delhi.

Kar, A. K. (2016). Bio inspired computing–A review of algorithms and scope of applications. Expert Systems with Applications, 59, 20-32.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

Karaboga, D., & Basturk, B. (2007a). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

Karaboga, D., & Basturk, B. (2007b, June). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In International Fuzzy Systems Association World Congress (pp. 789-798). Springer Berlin Heidelberg.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, 8(1), 687-697.

Kawala, F., Douzal-Chouakria, A., Gaussier, E., & Dimert, E. (2013, October). Prédictions d'activité dans les réseaux sociaux en ligne. In 4ième Conférence sur les Modèles et l'Analyse des Réseaux: Approches Mathématiques et Informatiques (p. 16).

Kiang, M. Y., Raghu, T. S., & Shang, K. H. M. (2000). Marketing on the Internet—who can benefit from an online marketing approach?. Decision Support Systems, 27(4), 383-393.

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., & Kavsek, B. (2000, August). Informal identification of outliers in medical data. In Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (Vol. 1, pp. 20-24).

Leskovec, J., Adamic, L. A., & Huberman, B. A. (2007). The dynamics of viral marketing. ACM Transactions on the Web (TWEB), 1(1), 5.

Malaga, R. A. (2008). Worst practices in search engine optimization. Communications of the ACM, 51(12), 147-150.

Mantegna, R. N., & Stanley, H. E. (1994). Stochastic process with ultraslow convergence to a Gaussian: the truncated Lévy flight. Physical Review Letters, 73(22), 2946.

Mills, A. J. (2012). Virality in social media: the SPIN framework. Journal of public affairs, 12(2), 162-169.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46-61.

Morales, A. J., Borondo, J., Losada, J. C., & Benito, R. M. (2014). Efficiency of human activity on information spreading on Twitter. Social Networks, 39, 1-11.

Mui, Y. Q., & Whoriskey, P. (2010). Facebook passes Google as most popular site on the Internet, two measures show. The Washington Post.

Murugavel, P., & Punithavalli, M. (2011). Improved hybrid clustering and distance-based technique for outlier removal. International Journal on Computer Science and Engineering (IJCSE), 3(1), 333-339.

Perlroth, N. (2013). Fake twitter followers become multimillion-dollar business. The New York Times, 5.

Ritson, M. (2013). Pay Up and Embrace Twitter's Fake Followers.Marketing Week, 46.

Roelens, I., Baecke, P., & Benoit, D. F. (2016). Identifying influencers in a social network: The value of real referral data. Decision Support Systems, 91, 25

Romero, D. M., Galuba, W., Asur, S., & Huberman, B. A. (2011, September). Influence and passivity in social media. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 18-33). Springer Berlin Heidelberg.

Slegg J. (2016). A Complete Guide to Panda, Penguin, and Hummingbird. Search Engine Journal.http://www.searchenginejournal.com/seo-guide/google-penguin-panda-hummingbird. accessed 15 February 2017.

Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012a, August). Integrating nature-inspired optimization algorithms to K-means clustering. In Digital Information Management (ICDIM), 2012 Seventh International Conference on (pp. 116-123). IEEE.

Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012b, August). Wolf search algorithm with ephemeral memory. In Digital Information Management (ICDIM), 2012 Seventh International Conference on (pp. 165-172). IEEE.

Taylor, C. (2012). Facebook Has 44% Fake Followers; Twitter Has 33%.

Tsugawa, S., & Ohsaki, H. (2017). On the relation between message sentiment and its virality on social media. Social Network Analysis and Mining, 7(1), 19.

Wang, G. G., Deb, S., Gandomi, A. H., Zhang, Z., & Alavi, A. H. (2016). Chaotic cuckoo search. Soft Computing, 20(9), 3349-3362.

Weng, L., Flammini, A., Vespignani, A., & Menczer, F. (2012). Competition among memes in a world with limited attention. Scientific reports, 2.

Weng, L., Menczer, F., & Ahn, Y. Y. (2013). Virality prediction and community structure in social networks. Scientific reports. 3.

Weng, L., Menczer, F., & Ahn, Y. Y. (2014). Predicting successful memes using network and community structure. Proceedings of 8th AAAI Intl. Conf. on Weblogs and social media (ICWSM 2014).

Yang, X. S. (2010a). Firefly algorithm, Levy flights and global optimization. In Research and development in intelligent systems XXVI (pp. 209-218). Springer London.

Yang, X. S. (2010b). A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), 65-74.

Yang, X. S. (2010c). A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), 65-74.

Yang, X. S. (2010d). Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation, 2(2), 78-84.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 210-214). IEEE.

Zhang, Y., Meratnia, N., & Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. IEEE Communications Surveys & Tutorials, 12(2), 159-170.

Zimek, A., Schubert, E., & Kriegel, H. P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining, 5(5), 363-387

PUBLICATIONS

Journals published online:

1. Aswani, R., Ghrera, S. P., & Chandra, S. (2016). A Novel Approach to Outlier Detection using Modified Grey Wolf Optimization and k-Nearest Neighbors Algorithm. Indian Journal of Science and Technology, 9(44). **[SCOPUS, Citation-3]**

2. Aswani, R., Ghrera, S. P., Kar, A. K., & Chandra, S. (2017). Identifying buzz in social media: a hybrid approach using artificial bee colony and k-nearest neighbors for outlier detection. Social Network Analysis and Mining, 7(1), 38. **[SCOPUS CiteScore 1.86, Emerging SCI, Springer, Citation-1]**

Conference/Book Chapters:

1. Aswani, R., Ghrera, S. P., Chandra, S., & Kar, A. K. (2017, November). Outlier Detection Among Influencer Blogs Based on off-Site Web Analytics Data. In Conference on e-Business, e-Services and e-Society (pp. 251-260). Springer, Cham. **[Springer LNCS CiteScore 0.67,SCOPUS, Citation-2]**

2. Aswani, R., Ghrera, S.P., Chandra, S., & Kar, A.K. (2017, December). Identifying popular online news: An approach using chaotic cuckoo search algorithm. Forthcoming in International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), IEEE **[IEEE, SCOPUS]**

3. Further, the first published journal article "A Novel Approach to Outlier Detection using Modified Grey Wolf Optimization and k-Nearest Neighbors Algorithm" was also presented at SHANNON 100 -THIRD INTERNATIONAL CONFERENCE held on 8th-9th April 2016 at Lovely Professional University, Jalandhar.

Journals under review:

1. Aswani, R., Ghrera, S. P., Kar, A. K., & Chandra, S. (2018). Identifying spam websites for search engine marketing using bio-inspired computing. Information Processing & Management **[Revise & Resubmit] [SCOPUS, SCI Expanded] [Elsevier]**

2. Aswani, R., Ghrera, S. P., Kar, A. K., & Chandra, S. (2018). Identifying fake profiles in social media marketing - Insights from Twitter Analytics. Future Generation Computer Systems **[Decision Pending] [SCOPUS, SCI Expanded] [Elsevier].**